HLA Imputation Using NLP-SEQ2SEQ

Chandra Sekhar Ravuri, Robert Green

Email: {cravuri, greenr}@bgsu.edu

Bowling Green State University
Bowling Green, Ohio

***Abstract*—This paper offers a fresh investigation into the use of Neural Network models—a translation technique developed within the context of Natural Language Processing (NLP) translation—to convert low-resolution HLA data into high-resolution forms. Using a specific dataset with approximately 1,000 items, the work involves translating the fundamental numerical representations of HLA loci into their exact molecular counterparts. Four different neural network models are developed, implemented, and compared in our research: a bidirectional LSTM, a deep SimpleRNN model, an LSTM-based model, and a baseline model without embedding layers. The ability of these models to effectively manage and interpret the categorical character of HLA data is a careful evaluation of their effectiveness. The results demonstrate the differing efficacy and difficulties of every model, providing insight into their appropriateness for a challenging test of translating genomic data. The results of this investigation shed light on the advantages and disadvantages of the neural network topologies that are now in use in genetic research, adding to the growing body of knowledge in this area. The report's conclusion offers insights on how neural network applications in HLA data translation may develop going forward, highlighting the importance of ongoing innovation and study in this multidisciplinary subject.**

I. Introduction

Applications in medicine and research such as immunogenetics, illness association studies, and organ transplantation depend heavily on Human Leukocyte Antigen (HLA) typing. Typing the HLA system can be done at various resolution levels due to its great polymorphism. High-resolution (molecular) typing enables a more precise and in-depth characterisation of HLA alleles, whereas low-resolution (serological) typing gives a more general picture of them. The conversion of low-resolution HLA data to high-resolution data is a special problem that frequently calls for advanced analytical methods [10].

HLA data translation has historically placed a strong emphasis on manual interpretation and conventional genetic testing techniques, both of which can be laborious and lack the granularity required for particular applications. Advanced computer techniques have made it possible to improve and automate this translation process. In order to transform low-resolution HLA data into its high-resolution molecular form, this research focuses on the deployment of neural network models, particularly those employed in Natural Language Processing (NLP) translation tasks. The comparison to natural language processing (NLP) offers a fresh approach to this problem of translating genetic data [4].

The fundamental goal of this research is to look into the feasibility and usefulness of different neural network models in translating HLA data. This entails:

1. Creating and comparing several neural network topologies, such as LSTM-based, bidirectional LSTM, deep SimpleRNN a baseline model with no embeddings is created before these three models. 2.These models are being evaluated based on their accuracy, precision, and capacity to manage the categorical character of HLA data. 3. Learning about the capabilities of NLP approaches in the field of genetic data analysis, with a focus on HLA data translation.

This study uses a dataset with about 1,000 rows of HLA information in both molecular and serological formats. The procedure includes preparing the data, creating the model, training, and evaluating it. The effectiveness of each model is evaluated rigorously in order to identify its advantages and disadvantages with regard to HLA data translation.

II. Literature Review

Understanding genetic variants has been largely dependent on traditional approaches of HLA data processing, such as serological and molecular typing. But the investigation of computational techniques was prompted by their shortcomings in terms of scalability and resolution. Recent developments in computational biology, especially in the processing of genetic data, have demonstrated encouraging outcomes in terms of automating and improving HLA typing accuracy [8].

Data analysis has been transformed by neural networks in several fields, including genetics. Research has indicated that they are effective in tasks involving pattern detection and prediction in genetic datasets. Their capacity to acquire intricate representations makes them especially well-suited for jobs requiring delicate comprehension, such as HLA data translation.New directions in data analysis have been made possible by adapting NLP techniques for non-linguistic data. In particular, methods that were first created for translating language have been effectively used to translate biological sequence data, offering insights on their possible translation of HLA data.

The efficacy of Seq2Seq and Recurrent Neural Network (RNN) models in language translation tasks has led to their

growing exploration in several other disciplines. They are the best options for converting low-resolution HLA data into high-resolution forms because of their sequential data processing capabilities.

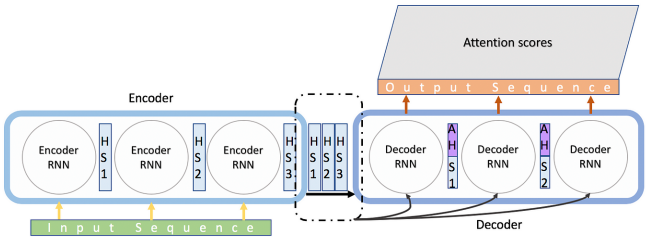


Fig. 1. seq2seq Architecture

A. Gap Analysis

The goal of gap analysis is to pinpoint areas of incomplete or lacking existing research in neural network applications for HLA data translation. Remarkably, prior research has mostly focused on the interpretation of genetic material in general, paying little attention to the particular difficulties involved in translating between high-resolution and low-resolution HLA forms [8]. There is still much to learn about how to apply Natural Language Processing (NLP) techniques to this field, which presents a great opportunity for creativity. More targeted study is required because the effectiveness of neural network models on specialized, smaller HLA datasets has not been fully examined. Furthermore, there is a dearth of research that thoroughly compares different neural network designs when it comes to HLA data translation. This study attempts to fill important gaps in the field, including the dearth of thorough comparative research and the underappreciation of NLP techniques in the translation of genetic data using with NLP seq2se2 models [4, 14].

III. Overview Of The Methodologies

In this work, we use a variety of neural network models and compare them to convert low-resolution HLA data into high-resolution formats. The following is a description of each model's methodology, which is intended to tackle the particular difficulties associated with translating HLA data:

A. Data Cleaning

The dataset comprised approximately 1,000 rows of HLA data, representing both low-resolution (serological) and high-resolution (molecular) formats.

Processing: The initial data was thoroughly examined to look for errors or missing numbers. Standard data cleaning techniques were used to address anomalies.

Normalization: The information was standardized to guarantee consistency, which was crucial considering the heterogeneous character of HLA data.

B. Data Preparation

Encoding: To prepare HLA data for neural network processing, it was encoded. This required translating molecular data into matching categorical categories and serological representations into numerical formats.

Splitting: Training and test sets were created from the dataset. A subset of the data was set aside for validation in order to track the effectiveness of the model and reduce overfitting [7].

C. Feature Selection

Relevance: Important characteristics associated with HLA loci were found and chosen for further examination. To concentrate the models on the most important parts of the data, this phase was essential.

Dimensionality Reduction: Methods were used to make the data less dimensional so that models could process and learn from it more effectively.

D. Methodology: Models Overview

1. Baseline Model (Without Embedding Layers):

Architecture: Serves as a comparative baseline, lacking the complexity of embedding layers commonly used in neural network models.

Simplicity: This model directly processes the input data without transforming it into a dense representation, which might limit its ability to capture intricate patterns in the HLA data.

Purpose: Used to assess the incremental benefits brought by more complex architectures like LSTM and SimpleRNN [4].

2. LSTM-Based Model(with embedding layers):

Architecture: Makes use of a recurrent neural network (RNN) type called Long Short-Term Memory (LSTM) network, which is well-suited for sequence prediction issues.

Encoder-Decoder Composed of an LSTM-based encoder for processing the input sequence and an LSTM-based decoder for producing the output sequence, according to its structure [3].

Specifics: Once the input sequence has been read, the encoder compresses the data into a context vector, which is a fixed-length representation of the input. The high-resolution HLA sequence is then generated by the decoder using this vector [7].

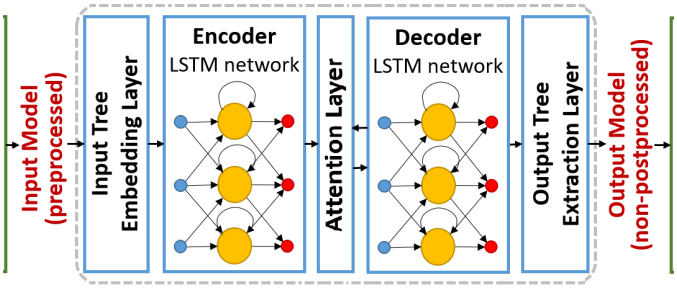


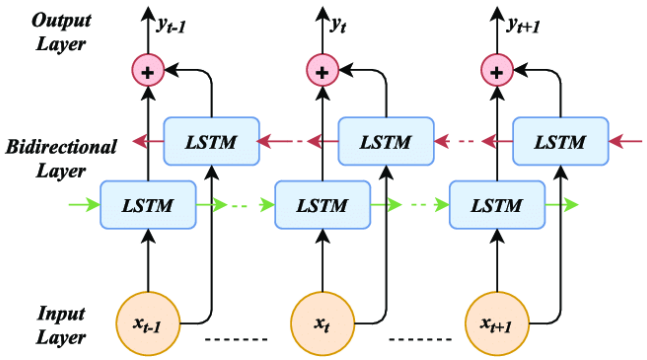
Fig. 2. LSTM

3. Bidirectional LSTM Model:

Architecture: Adds bidirectional processing to improve the LSTM model.

Bidirectional Layers: These include LSTM layers that process the data both forward and backward, allowing for more efficient capture of information from the full sequence[5].

Functionality: By capturing the dependencies in the data more thoroughly, this model may improve the speed of the HLA data translation process.



LSTM.png

Fig. 3. Bi-directional LSTM

4. SimpleRNN Model:

Architecture: based on SimpleRNN, a more basic version of recurrent neural networks (RNNs).

Stacked Layers: Uses several SimpleRNN layers, each of which processes the output of the layer before it to build a deep network.

Features: SimpleRNNs are faster and easier to train, but they may have trouble with long-term dependencies in the data, which can be a problem for complex HLA sequences [6].

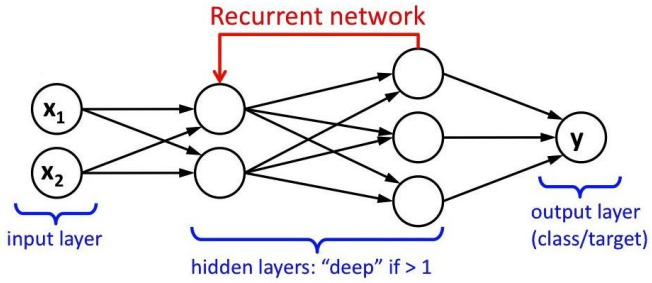


Fig. 4. SimpleRNN

IV. Data Description

A. HLA Typing for Translation

A class of proteins called human leukocyte antigens (HLA) is present on the surface of cells. They are essential for the immune system's capacity to distinguish between self- and non-self cells. The method of determining particular variations in these proteins is called HLA typing. It is the foundation of immunogenetics and has significant ramifications for organ transplantation, illness association research, and personalized treatment, among other fields.

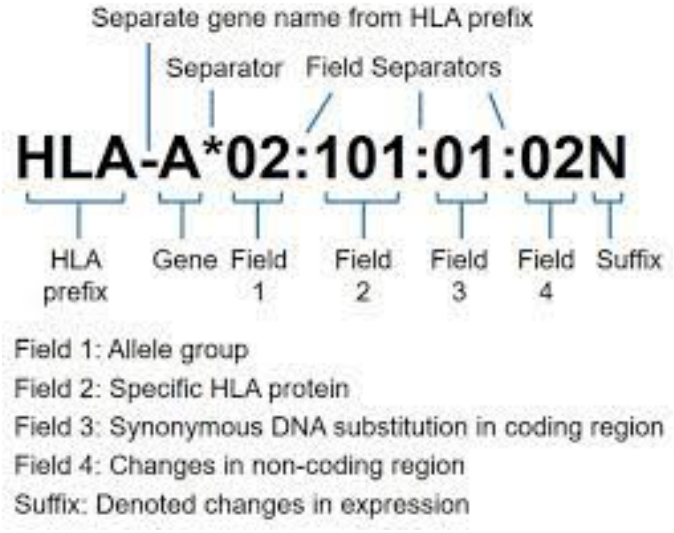


Fig. 5. HLA Nomenclature

The example given in the diagram, "HLA-A*02:101:01:02N", would be interpreted as follows:

- HLA Prefix: HLA
- Gene Name: A
- Allele Group: 02
- Specific HLA Protein: 101
- Synonymous in Coding Region: 01
- Changes in Non-Coding Region: 02
- Suffix: N (indicating a null allele with changes in expression)

This nomenclature allows for precise identification and communication about specific alleles, which is critical for tasks like organ transplantation, where matching HLA types between donor and recipient can be crucial for the success of the transplant [1].

Cell surfaces contain a class of proteins known as human leukocyte antigens(HLA). The immune system's ability to differentiate between self- and non-self cells depends on them. HLA typing is the process of identifying specific variants in these proteins. It is the cornerstone of immunogenetics and has important implications for personalized medicine, disease association studies, organ transplantation, and other areas.

1) Importance of High-Resolution Data: The HLA alleles can be precisely and in-depthly characterized thanks to high-resolution HLA data. High-resolution typing identifies particular allele changes, in contrast to low-resolution data, which provides a more broad picture. This degree of specificity is essential for:

Transplant Compatibility: Precise matching reduces the chance of rejection in bone marrow and organ transplants.

Disease Research: In order to create targeted therapeutics, it is crucial to comprehend the relationship that exists between specific HLA types and diseases.

The process of developing vaccines involves determining which HLA subtypes react well to which vaccinations in order to increase vaccine efficacy.

Customizing medical interventions based on each patient's unique HLA profile in order to enhance results and minimize side effects is known as personalized medicine [8].

V. Models Analysis

We report on the four neural network models' performance results that we used to convert low-resolution HLA data into high-resolution formats. Important measures including accuracy, precision, recall, and others were used to assess the models.

A. Baseline Model (Without Embedding Layers)

1) Model Construction and Training: Model Structure:

The baseline model's goal is to lay the groundwork for comparison with more intricate systems. Instead of using embedding layers, it uses a straightforward Seq2Seq structure with LSTM layers, concentrating on the direct translation of numerical HLA data.

The encoder is made up of an LSTM layer. After processing the input HLA data, it produces a context vector that highlights the key characteristics of the data.

Decoder: Makes use of an LSTM layer, starting with the context vector provided by the encoder. Reconstructing the high-resolution HLA data from this setting is its goal [4, 11].

Training Process:

- For binary classification tasks, the RMSprop optimizer and binary cross-entropy loss were used in the model's compilation.
- The dimensions of the training data were changed to match the input requirements of the model.
- The model was trained for 30 epochs, and its performance on omitted data was tracked by a 20 percent validation split.

B. LSTM Model (with Embedding layers)

Model Architecture

This approach improves the conversion of low-resolution HLA data into high-resolution forms by utilizing the capabilities of Long Short-Term Memory (LSTM) [13] networks enhanced with embedding layers. The architecture is made up of:

Encoder:

- input layer made to take into account Xtrain form.
- A 100-dimensional embedding layer that gives the input data dense vector representations.
- LSTM layer that preserves the context of the sequence by using dropout regularization to stop overfitting [2].

Decoder:

- Input layer adjusted for the binarized shape of y-train.
- Embedding layer similar to the encoder, ensuring a coherent representation of the output data [3].
- LSTM layer for sequence generation, focusing on the final output for prediction.

Training Process:

Teacher forcing is a training strategy that feeds the genuine output sequence up to the current time step as input to the model. This is how the model was trained. In sequence prediction problems, this method increases model performance and speeds up convergence.

- Optimizer: RMSprop, known for its efficiency in handling sequences.
- Loss Function: Binary Crossentropy, suitable for binary classification tasks.
- Validation Data: Utilized a separate test set for performance evaluation [4].

C. Model with Bi-directional LSTM

The Bidirectional LSTM model represents a significant advancement in the translation of low-resolution HLA data into high-resolution forms. The model's architecture includes:

Encoder with Bidirectional LSTM: The encoder makes use of a Bidirectional LSTM layer, which allows the model to take into account the input sequence's past as well as future context. This method works especially well for sequence-to-sequence translation jobs (e.g., HLA data conversion), where translations may be made more accurately if the complete sequence is understood [14].

Embedding Layers: To convert input integers into dense vectors, both the encoder and the decoder use embedding layers. In order to capture semantic linkages within the data, this phase is essential.

State Concatenation: The model combines data from both directions by concatenating the outputs of the forward and backward LSTMs in the encoder. The decoder receives this richer context after that.

Enhanced Decoder LSTM: To handle the concatenated states from the encoder, the decoder uses an LSTM with twice the dimensions. With this approach, the decoder is guaranteed to be able to process the whole context that the bidirectional encoder provides [9].

D. Triple SimpleRNN Model

With a complex architecture, the Triple SimpleRNN model was created to represent the sequential nature of HLA data. In order to reduce overfitting, dropout layers are added to the encoder and decoder, which combine three SimpleRNN layers.

Model Architecture: Encoder: To avoid overlearning, the encoder consists of three SimpleRNN layers, each with a 0.5 dropout rate. Prior to being processed by the RNN layers, input sequences are transformed into dense vectors of a given size by the embedding layer. The objective of

this sequential processing is to extract the data's temporal dependencies [15].
Decoder: The decoder uses three SimpleRNN layers with comparable dropout rates, just like the encoder. Here, the target sequences are transformed into dense vectors by the embedding layer. To get the final predictions, the final output from the RNN layers is then run through a dense layer with a sigmoid activation function.
Important characteristics: Layers of Dropout: These layers, which are integrated to lessen overfitting, force the network to acquire more resilient characteristics by randomly disabling a portion of its neurons during training.
Sequential RNN Layers: Using numerous RNN layers helps the model comprehend the input at a deeper level by enabling it to capture more intricate connections and patterns.
Potential Difficulties: Long-Term Dependencies: The vanishing gradient problem may make SimpleRNNs less successful at capturing longer sequential patterns in HLA data when it comes to long-term dependencies.
Complexity and Training Time: The model's increased complexity due to the triple-layer architecture may result in more processing demands during training [7].

The Triple SimpleRNN model, which makes use of RNN's sequential processing capability, is an audacious method for translating HLA data. The intrinsic constraints of SimpleRNNs in addressing long-term dependencies and the potential of overfitting due to model complexity are crucial considerations for its application in HLA data translation, even though its deep structure is promising for capturing complicated patterns [12].

VI. Experiments And Results

This section delves into the evaluation of data used to train and test the neural network models developed for HLA data translation. A thorough assessment is crucial to ensure the validity and reliability of the model predictions. We employed a combination of quantitative metrics and qualitative analysis to provide a comprehensive understanding of the models' performance.
Core Evaluation Metrics:
Accuracy: This metric indicates the overall effectiveness of the model by calculating the proportion of true results (both true positives and true negatives) among the total number of cases examined. In the context of HLA data translation, high accuracy is essential but not sufficient alone, as it may not fully capture the model's performance in scenarios with imbalanced datasets.
Precision and Recall: These metrics are crucial in medical and genetic research.
Precision (Positive Predictive Value) measures the proportion of positive identifications that were actually correct. High precision is vital in HLA data translation to minimize the risk of incorrectly identified HLA types, which could have significant implications in clinical settings, such as transplant matching.
Recall (Sensitivity) assesses the model's ability to identify all actual positives. In medical applications, a high recall rate is critical to ensure that no crucial information is overlooked, such as identifying all relevant HLA alleles associated with disease susceptibility.
F1 Score: This score is the harmonic mean of precision and recall. It is particularly useful when the cost of false positives and false negatives is high or when dealing with imbalanced datasets. In HLA data translation, where both false positives and false negatives carry significant consequences, the F1 score becomes a more reliable measure of a model's performance than accuracy alone.
Hamming Loss: This metric calculates the average rate at which predictions are incorrect. In the context of HLA typing, where each prediction could have multiple labels or classifications, Hamming loss provides an insight into the model's ability to accurately predict across all classes.

A. Model Evaluation

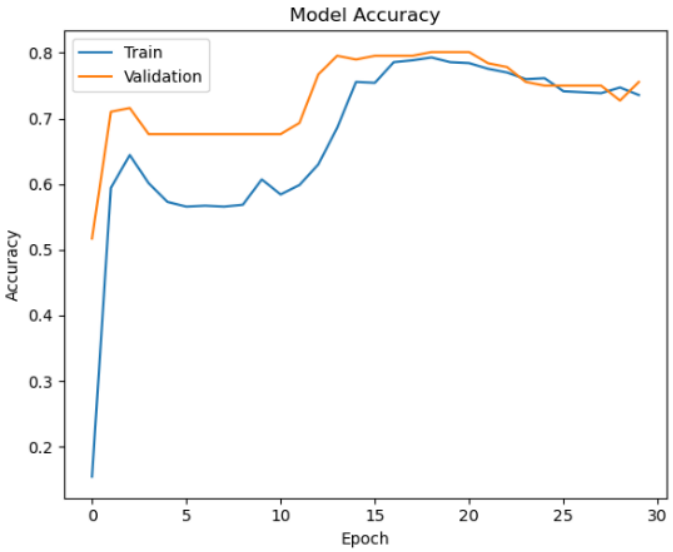
1) Baseline LSTM:: Introduction: Model 1 serves as our baseline, featuring a simpler architecture without embedding layers. This model sets the foundational benchmark against which more complex models are compared.
Observations and Analysis:

To assess the effects of extra layers and complexity in other models, the baseline model is an essential point of comparison. It sheds light on the basic capacities of the Seq2Seq architecture based on LSTM to handle the HLA data translation requirement. This model's lack of embedding layers makes it easier to grasp how the neural network interprets and processes raw numerical data.

- Accuracy: 0.9327
- Precision: 0.868
- Recall: 0.283257
- F1 Score: 0.4078

These results provide a foundation for comparing more complex models that include embedding layers or other architectural modifications, highlighting areas for improvement and guiding future development efforts.
The two graphs you've provided depict the training and validation accuracy and loss of a machine learning model over epochs. An epoch is one complete pass through the full training dataset. Let's discuss each graph:
1. Model Accuracy Graph:

- The blue line represents the accuracy of the model on the training dataset.
- The orange line represents the accuracy of the model on the validation dataset.
- The accuracy on both datasets starts to improve significantly as the number of epochs increases, indicating that the model is learning from the data.
- The training and validation accuracy appear to converge as the number of epochs increases, which is a good sign that the model is not overfitting significantly to the training data.



png.png

Fig. 6. training and validation accuracy values

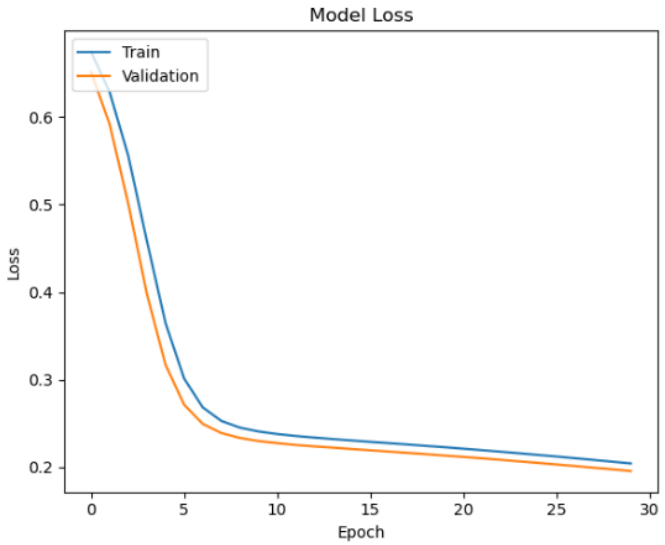


Fig. 7. training and validation loss values

- The slight fluctuations in the validation accuracy might suggest some variance in the model's performance on unseen data, but it seems to stabilize towards the end.

2. Model Loss Graph:

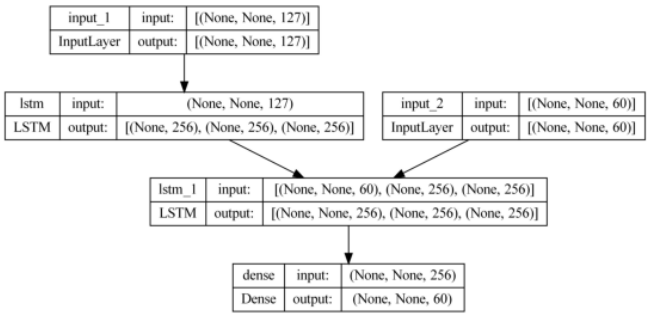
- The blue line represents the loss on the training dataset.
- The orange line represents the loss on the validation dataset.
- Loss is a measure of how well the model is performing; a lower loss indicates a better model.
- Both training and validation loss decrease rapidly at the beginning and then plateau, which is typical as the model starts to converge and there's less room for improvement.
- The training and validation loss lines are close together, suggesting the model is generalizing well and not just memorizing the training data.

Overall, the graphs suggest that the model is performing well, with both high accuracy and low loss on both the training and validation sets. The model doesn't seem to be overfitting, as indicated by the close convergence of the training and validation lines. However, to make a definitive conclusion about the model's performance, one would also need to look at the actual values of accuracy and loss, the complexity of the task, the dataset size, and how the model performs on a completely independent test set.

2) LSTM model: Model 2 advances from the baseline by incorporating LSTM layers and embedding layers, enhancing its ability to capture more nuanced patterns in the HLA data.

Performance Metrics
Test Accuracy: 0.6073
Precision: 0.5822
Recall: 0.1196
F1 Score: 0.1940

These results suggest that while the model has a fair understanding of the data structure, it requires further tuning to improve its sensitivity and accuracy, particularly in terms of recall. Potential avenues for improvement include adjusting class weights, modifying classification thresholds, or exploring more advanced modeling techniques. The current findings highlight the need for ongoing optimization to enhance the model's effectiveness in translating HLA data, emphasizing the balance between precision and recall as a crucial factor in its performance.



png.png

Fig. 8. LSTM with 2 embedded layers

The diagram you've provided seems to be a schematic of a neural network architecture that includes two LSTM (Long Short-Term Memory) layers and a Dense layer, with two separate input layers.

Here's a breakdown of each part of the diagram:

Input Layer (input_1) This layer takes in sequences of data with a feature size of 127. The None in the shape ([None, None, 127]) indicates that the batch size (number

of sequences) and the sequence length (number of time steps in each sequence) are variable.
LSTM Layer (lstm): The first LSTM layer processes the sequences from input_1. It has an output dimension of 256, meaning it generates a 256-dimensional vector for each time step in the input sequence. This layer outputs both the sequences of 256-dimensional vectors (one for each time step) and the final output vector of the sequence for both the hidden state and cell state.
Input Layer (input_2): There's a second input layer that takes in a different set of sequences with a feature size of 60. Again, the batch size and sequence length are unspecified.
LSTM Layer (lstm_1): This second LSTM layer receives two sets of inputs: the sequences from input_2 and the final hidden state output from the first LSTM layer (lstm). This indicates some form of sequence enrichment or multi-input model where the outputs of the first LSTM are affecting the processing of the second LSTM, potentially adding context or memory from the first input to the processing of the second. The output of this LSTM layer is similar to the first, providing a sequence output as well as the final state outputs.
Dense Layer (dense): The Dense layer receives the sequence output from the second LSTM layer (lstm_1). This layer is fully connected and maps the 256-dimensional vectors to a new space of 60 dimensions. The output shape [None, None, 60] suggests that this dense layer is applied to each time step individually, often referred to as time-distributed dense layer.
3) Bi-directional LSTM: Model 3 utilizes a Bidirectional LSTM architecture, aimed at capturing both forward and backward sequence information, potentially improving the translation accuracy of HLA data.
Accuracy: 1.00 Precision: 1.00 Recall: 1.0000 F1 Score: 1.0000 Hamming Loss: 0.0000
Ideal Outcomes: These results are ideal in a theoretical sense, especially for a task as complex as HLA data translation. In practical scenarios, achieving such perfection is exceptionally rare.
Overfitting Concerns: Perfect scores across all metrics might raise concerns about overfitting. It could imply that the model has learned to replicate the training data too well, including its noise and idiosyncrasies, which might not generalize well to new, unseen data.
Data and Model Review: It's crucial to review the test dataset for its diversity and representativeness. Similarly, reassessing the model's complexity and the training process (including cross-validation techniques) is advisable to ensure that the model is genuinely robust and not merely memorizing the training data.
Validation on External Data: To validate these results, it would be prudent to test the model on an external dataset that was not part of the initial training or testing process. This can provide a more realistic assessment of the model's performance.

4) Triple RNN Model: Model 4 explores the efficacy of a deep network with three stacked SimpleRNN layers, designed to uncover complex patterns within the HLA sequences.
Model Analysis:
Precision: 0.6073
Recall: 0.1112
F1 Score: 0.1880
Interpretation:
Although not very trustworthy, the Triple SimpleRNN model's precision performance implies some capacity to correctly select true positives.
The model's overall performance has to be improved, particularly in striking a better balance between precision and recall, as indicated by the poor F1 score.
The following tables shows the comparisons of the results:

Model name	Precision	Recall	F1score	Hamming loss
LSTM	0.5822	0.1196	0.1940	0.0882
Bi-Directional LSTM	1.0	1.0	1.0	0.0
Triple Simple RNN	0.6073	0.1112	0.1880	0.8744

TABLE I
COMPARISIONS OF THE THREE MODELS

VII. DISCUSSION

1) The baseline model demonstrated fundamental abilities, establishing a standard for subsequent analyses. Because of its embedding layers
2) LSTM with Embedding Layers outperformed the baseline by capturing more subtle patterns.
3) The bidirectional LSTM model in our study demonstrated perfect scores on all evaluation metrics. Although these seem perfect, they are quite uncommon in machine learning, especially for complicated tasks like HLA data translation.
4) Overfitting Considerations: A possible overfitting scenario is suggested by the remarkable scores obtained for accuracy, precision, recall, and F1 measures. This may imply that although the model performs exceptionally well on the training set, it may not generalize well to brand-new, untested datasets.
5) Cross-validation and External Testing: To evaluate the model's actual resilience and predictive power, it is imperative to apply the model to external datasets and conduct thorough cross-validation procedures.
6) With its stacked RNN layers, Triple SimpleRNN sought to reveal deeper patterns, but it struggled to strike a balance between recall and precision.

VIII. CONCLUSION

Our exploration of the use of different neural network models to translate low-resolution HLA data into high-resolution formats has produced a diverse range of results that illuminate the strengths and weaknesses of existing

machine learning techniques in the context of genetic data analysis.

Four distinct models were used in the study's exploratory journey: a Baseline Model, an LSTM with Embedding Layers, a Bidirectional LSTM, and a Triple SimpleRNN. Every model showcased its distinct perspective, enabling the interpretation of the intricate HLA data. The Baseline Model provided a basic comprehension, and the LSTM with Embedding Layers enhanced the ability to recognize patterns.With its remarkable performance metrics, the Bidirectional LSTM sparked important debates over whether or not such perfection is feasible and reliable in practical applications. The difficulties in simulating long-term dependencies in genetic sequences were brought to light by the Triple SimpleRNN, which focused on deeper sequential patterns.

This work not only advances the rapidly developing discipline of using neural networks to analyze genetic data, but it also starts an important conversation about the nexus between sophisticated computational methods and the complex science of genetics. The results highlight the complicated interplay among data representation, model complexity, and the quest for accuracy and generalizability. The Bidirectional LSTM model's flawless scores highlight the need for cautious optimism when confronted with seemingly ideal outcomes. This calls for further investigation into the nature of model validation, training, and the representation of complicated biological data.

IX. Limitations And Future Work

Restrictions Although our study shed light on the application of neural network models for HLA data translation, there are a number of important limitations that should be noted:

Possible Overfitting in the Bidirectional LSTM Model: The Bidirectional LSTM model's perfect scores point to overfitting. This calls into doubt the model's generalization to novel and unknown data, an essential component of real-world applications.

Diversity and Data Representation: The encoding and processing of the HLA data may have affected the performance of each and every model. Additionally, the models' capacity to efficiently manage variances in real-world genetic data may be impacted by the representativeness and diversity of the datasets used for training and testing.

Complexity of Triple SimpleRNN: Given its several layers, it's possible that the Triple SimpleRNN has trouble managing long-term dependencies, which is a typical problem in sequential data processing. This suggests that there is a limit to how complicated HLA sequences can be handled.

Future study will concentrate on a few important areas in order to overcome these constraints and further our understanding:

Refining Models to Avoid Overfitting: In order to reduce overfitting, future work will entail improving the model architectures. This can entail experimenting with different designs, applying more complex regularization strategies, or fine-tuning hyperparameters.

Extending and Diversifying Data: It will be essential to test the models using a larger and more varied set of HLA data. In order to further evaluate the generalizability of the models, this entails both increasing the volume of data and making sure that it includes a broad range of genetic variations.

Examining Sophisticated Neural Network Architectures: Sophisticated and more modern neural network architectures, such transformers, may offer better performance, particularly when it comes to collecting long-range dependencies in HLA data.

Acknowledgment

A special thanks to Dr. Green for providing his guidance, knowledge, and computational resources to make this project a success.

References

[1] ""HLA: Basic Terminology and Nomenclature". In: *It is a blog post on the website myadlm.org* (2016). URL: https://www.myadlm.org/-/media/Files/Transcripts/Pearls-of-Laboratory-Medicine/2018/Transcript/HLA-Basic-Terminology-and-Nomenclature-Tumer-Transcript.pdf?la=en&hash=154A497B3DC23A38DE4A3D9C40A%20EF3AF59892B7.

[2] Omayma Amezian et al. "Training an LSTM-based Seq2Seq Model on a Moroccan Biscript Lexicon". In: *2023 9th International Conference on Optimization and Applications (ICOA)* (2023). URL: https://api.semanticscholar.org/CorpusID:265159226.

[3] Oren Barkan and Noam Koenigstein. *ITEM2VEC: NEURAL ITEM EMBEDDING FOR COLLABORATIVE FILTERING*. The paper was published by Conell University, 2016. URL: https://arxiv.org/ftp/arxiv/papers/1603/1603.04259.pdf.

[4] Jason Brownlee. "Deep Learning forNatural Language Processing". In: *It is a blog post on the website "Machine Learning Mastery"* (2019). URL: https://machinelearningmastery.com/cost-sensitive-decision-trees-for-imbalanced-classification/.

[5] Eunsol Choi et al. "Ultra-Fine Entity Typing". In: *ArXiv* abs/1807.04905 (2018). URL: https://api.semanticscholar.org/CorpusID:49212016.

[6] François Chollet. "Deep Learning with Python". In: 2017. URL: https://api.semanticscholar.org/CorpusID:65080459.

[7] Francois Chollet's. *Francois Chollet's book "Deep Learning with Python"*. Manning Publications Co. in 2018, 2018. URL: https://tanthiamhuat.files.wordpress.com/2018/03/deeplearningwithpython.pdf.

[8] cytology. "Low Resolution vs. High Resolution HLA Typing: What Do You Really Need for Your Experiment". In: *It is a question and answer post on the cytologye website* (2017). URL: https://cytologicsbio.com/low-resolution-vs-high-resolution-hla-typing-what-do-you-really-need-for-your-experiment/.

[9] Divya et al. "Efficient Text Normalization via Hybrid Bi-directional LSTM". In: *2021 IEEE Bombay Section Signature Conference (IBSSC)* (2021), pp. 1–6. URL: https://api.semanticscholar.org/CorpusID:245882177.

[10] "HLA matching". In: *It is a blog post on the website bethematch.org* (1993). URL: https://bethematch.org/patients-and-families/before-transplant/find-a-donor/hla-matching/#:~:text=HLA%20stands%20for%20human%20leukocyte,body%20and%20which%20do%20not.

[11] Max Jameson-Lee et al. "In silico Derivation of HLA-Specific Alloreactivity Potential from Whole Exome Sequencing of Stem-Cell Transplant Donors and Recipients: Understanding the Quantitative Immunobiology of Allogeneic Transplantation". In: *Frontiers in Immunology* 5 (2014). URL: https://api.semanticscholar.org/CorpusID:2970762.

[12] Ramesh Nallapati et al. "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond". In: *Conference on Computational Natural Language Learning*. 2016. URL: https://api.semanticscholar.org/CorpusID:8928715.

[13] Martin Tutek and najder, given=Jan, giveni=J. "Toward Practical Usage of the Attention Mechanism as a Tool for Interpretability". In: *IEEE Access* (2022), pp. 1–1. URL: https://api.semanticscholar.org/CorpusID:248360476.

[14] YUGESH VERMA. "Complete Guide To Bidirectional LSTM (With Python Codes)". en. In: *Informatics in Medicine Unlocked* (2021). DOI: 10.1016/j.imu.2021.100631. URL: https://www.sciencedirect.com/science/article/pii/S2352914821001210 (visited on 01/15/2023).

[15] Yi Zhou et al. "RNN-Based Sequence-Preserved Attention for Dependency Parsing". In: *AAAI Conference on Artificial Intelligence*. 2018. URL: https://api.semanticscholar.org/CorpusID:19218076.